



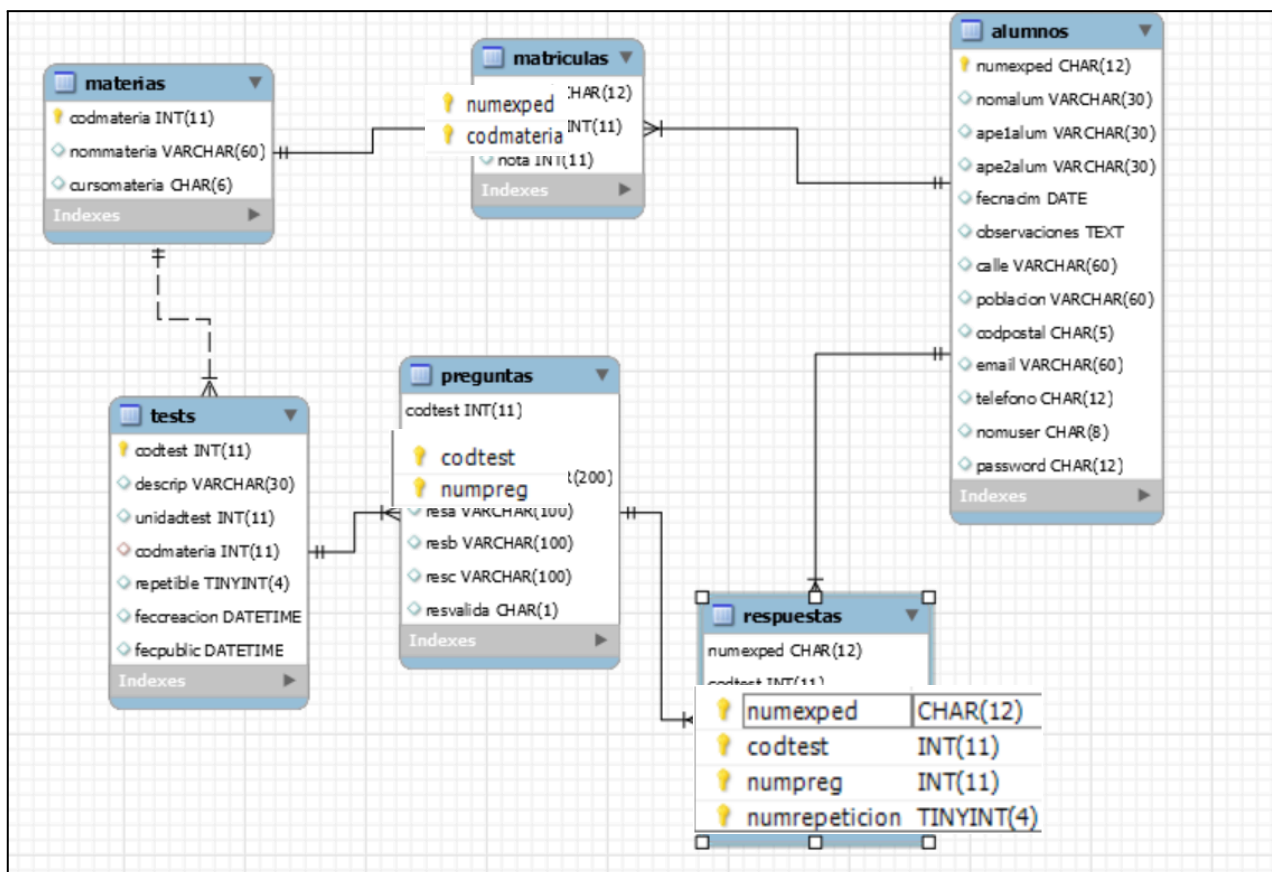
Puntuación total: 10

Examen - Unidad 8. Elementos Avanzados de BBDD Relacionales

Nombre de alumno/a:

Fecha:

Para el siguiente esquema relacional que pertenece a la base de datos de un SI de gestión de tests online de un centro académico:



- T.1.** Explica mediante una tabla de procesos la ejecución del siguiente procedimiento (VER DETRÁS) en el caso de que dos usuarios accedan a la vez al procedimiento.
- T.2.** Para el ejemplo anterior, añade lo que consideres oportuno para que se garantice el acceso adecuado a la base de datos por más de un usuario a la vez y/o para que en caso de producirse un error inesperado, se deshiciera todo. Explica razonadamente para este caso, según lo que has obtenido en la tabla de procesos de T1, para que es necesario añadir el código que has decidido añadir en este ejercicio.
- P.1.** Se quiere asegurar que no se repiten preguntas en el mismo test. Esto es, si se añade una pregunta a un test, si esta ya existía para ese mismo test, no se debe añadir. En el caso de que se modifique una pregunta, si se está sustituyendo por una existente, no se hará la modificación aunque esto será transparente al usuario.
- P.2.** Se ha detectado que en nuestro sistema se hacen multitud de consultas buscando tests por código de materia y unidad y queremos asegurar que estas consultas se hacen lo más ágilmente posible. Una vez hecho lo que consideres necesario para garantizar ésto, explica razonadamente qué sucederá cuando se hagan operaciones de manipulación de tests.



Puntuación total: 10

Examen - Unidad 8. Elementos Avanzados de BBDD Relacionales

Nombre de alumno/a:

Fecha:

P.3. Se ha decidido que para el curso actual, que comienza en septiembre y termina en junio, al finalizar cada trimestre se va a subir un punto en la nota del alumnado en las materias en las que haya hecho más de 10 tests.

P.4. Queremos asegurarnos que no vamos a tener dos posibles respuestas iguales en una pregunta, esto es, las tres respuestas posibles de una pregunta deben ser diferentes.

P.5. (Extra) Añade el código que falta en el procedimiento para añadir matrículas del alumnado.

```
/** PROCEDIMIENTO PARA AÑADIR LAS MATRÍCULAS DEL ALUMNADO */  
use gbdgestionatests;
```

```
-- NOTA: Para poder hacer este ejercicio, debes añadir una columna en la tabla matrículas:
```

```
-- (este campo valdrá 0 la primera vez que un alumno/a curse una materia y 1, 2, etc. cuando repita)
```

```
alter table matriculas  
add column repite tinyint default 0
```

```
delimiter $$
```

```
create procedure matriculaAlumnado
```

```
(in numeroexped char(12),
```

```
in nombrealum varchar(30),
```

```
in apellido1alum varchar(30),
```

```
in apellido2alum varchar(30),
```

```
in fechanacimalum date,
```

```
in callealum varchar(60),
```

```
in poblacionalum varchar(60),
```

```
in codpostalalum char(5),
```

```
in emailalum varchar(60),
```

```
in telefonoalum char(12),
```

```
in nomuseralum char(8),
```

```
in passwordalum char(12),
```

```
in cursomatricula char(6)
```

```
)
```

```
begin
```

```
/*Zona de declaración.*/
```

```
declare alumnomatriculado boolean default false;
```

```
declare numrepeticion tinyint default 0;
```

```
/*Zona de código.*/
```

```
/* A. Buscamos si el alumno/a ha estado dado de alta anteriormente */
```

```
set alumnomatriculado = exists (select * from alumnos where numexped = numeroexped);
```

```
/* B. Si el alumno estaba matriculado anteriormente, nos aseguramos que sus datos sean los que nos han pasado,
```

```
si no, insertaremos al alumno */
```

```
if alumnomatriculado then
```

```
update alumnos
```

```
set nomalum = nombrealum,
```

```
ape1alum = apellido1alum,
```

```
ape2alum = apellido2alum,
```

```
fechnacim = fechanacimalum,
```

```
calle = callealum,
```

```
poblacion = poblacionalum,
```

```
codpostal = codpostalalum,
```

```
email = emailalum,
```

```
telefono = telefonoalum,
```



Puntuación total: 10

Examen - Unidad 8. Elementos Avanzados de BBDD Relacionales

Nombre de alumno/a:

Fecha:

```

nomuser = nomuseralum,
password = passwordalum
where numexped = numeroexped;
else
insert into alumnos
(numexped, nomalum, ape1alum, ape2alum, fecnacim, calle, poblacion, codpostal, email, telefono,
nomuser,password)
values
(numeroexped,nombrealum,apellido1alum,apellido2alum,fechanacimalum,callealum,poblacionalum,codpostalalum,emailalum,telefon
oalum,nomuseralum,passwordalum);
end if;

/* C. Si el alumno estaba matriculado, comprobamos el número de veces de repetición de un curso */
if alumnomatriculado then
set numrepeticion = (.....) /* PUNTO EXTRA */
else
set numrepeticion = 0;
end if;

/* D. Añadimos las matrículas */
..... /* PUNTO EXTRA */

end $$
delimiter ;

```

TABLA DE PROCESOS T1 (VERSIÓN 1)									
DATOS ANTES DE COMENZAR:		Usuario 1 ⇒ matrícula del expediente '00100' en el curso '1ESO' por primera vez Usuario 2 ⇒ matrícula del expediente '00200' en el curso '1ESO' por segunda vez							
TIEMPO:	T1	T2	T3	T4	T5	T6	T7	T8	T9
Usuario 1	A	B	C	D					
DATOS/ SITUACIÓN	alumnomatriculado = false	insert alumno 00100	numrepeticion = 0	insercción matrícula alumno 00100					
Usuario 2	espera	A	B	C	D				
DATOS/ SITUACIÓN		alumnomatriculado = true	actualización datos alumno 00200	numrepeticion = 1	insercción matrícula alumno 00200				

TABLA DE PROCESOS T1 (VERSIÓN 2 - MÁS CORRECTA)



Puntuación total: 10

Examen - Unidad 8. Elementos Avanzados de BBDD Relacionales

Nombre de alumno/a:

Fecha:

DATOS ANTES DE COMENZAR:		Usuario 1 ⇒ matrícula del expediente '00100' en el curso '1ESO' por primera vez Usuario 2 ⇒ matrícula del expediente '00200' en el curso '1ESO' por segunda vez							
TIEMPO:	T1	T2	T3	T4	T5	T6	T7	T8	T9
Usuario 1	A	B	C	D					
DATOS/ SITUACIÓN	alumnomatriculado = false	insert alumno 00100	numrepeticion = 0	insercción matricula alumno 00100					
Usuario 2	A	B	B	C	D				
DATOS/ SITUACIÓN	alumnomatriculado = true	en espera	actualización datos alumno 00200	numrepeticion = 1	insercción matricula alumno 00200				

Razonamiento T2 ⇒ la transacción sirve en este caso solo para que si se produce una situación inesperada dentro del procedimiento, se deshaga todas las operaciones, pero no sirve para evitar errores, ya que, en este ejemplo, no se producirán errores si los dos usuarios ejecutan el procedimiento a la vez, esto es porque las claves primarias son diferentes y no dependen del resultado de una operación dentro del procedimiento..